UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/977,081	10/12/2001	Weiping Li	WCT-7031	1912

7590   11/03/2004

Martin Novack
17414 Via Capri East
Boca Raton, FL 33496

RECEIVED
NOV 2 2 2004
Technology Center 2600

EXAMINER
LE, VU

ART UNIT	PAPER NUMBER
2613	

DATE MAILED: 11/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO 90C (Rev 10/03)

Supplemental ● Office Action Summary	Application No. ● 09/977,081	Applicant(s) LI, WEIPING	
	Examiner Vu Le	Art Unit 2613	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL.**   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.

	EXAMINER

ART UNIT	PAPER
	20041027

DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner for Patents

A Supplemental Office Action to the one mailed September 24, 2004 is attached herewith with a period of response restarted 3 months from the date of mailing of this Supplemental Action.

Primary Examiner
Art Unit: 2613

09/977081

SERIAL NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.

EXAMINER
VU LE

ART UNIT	PAPER NUMBER
2613	

DATE MAILED:

Please find below a communication from the EXAMINER in charge of this application.

Commissioner of Patents and Trademarks

Responsive to Communication Filed _September 27, 2004._

The enclosed is a correct copy of a reference relating to the last Office action. The correction is indicated below.

THE PERIOD FOR RESPONSE OF _____3_____ MONTHS SET IN SAID OFFICE ACTION IS RESTARTED TO BEGIN WITH THE DATE OF THIS LETTER.

Part 1 - ~~Correct Reference Citation~~ Supplemental Office Action + PTO-892 resubmitted with corrected Correspondence address.

by _____ VU LE
~~PRIMARY EXAMINER~~

☐ Part 2 - Correct Reference Furnished:

_____

_____

by _____
Reference Order Center

enc.

PTOL-316 (rev. 6/78)

DETAILED ACTION

Claim Rejections - 35 USC § 102

1.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in-

(1) an application for patent, published under section 122(b), by another filed in the United States before the invention
by the applicant for patent; or
(2) a patent granted on an application for patent by another filed in the United States before the invention by the
applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have
the effects for the purposes of this subsection of an application filed in the United States only if the international
application designated the United States and was published under Article 21(2) of such treaty in the English.

2.   Claim 1, 3, 5-7, 9-11 are rejected under 35 U.S.C. 102(e) as being anticipated by

van der Schaar et al, US 2002/0006161.

Re claim 1, van der Schaar et al discloses a method for encoding frames of input

video (fig. 3a), comprising the steps of:

processing said input video ("original video", 106) to produce a compressed base

layer bit stream (110);

processing said input video to produce a compressed enhancement layer bit

stream (150⁵);

identifying a region of interest in a video frame (section 0023);

and enhancing the quality of the region of interest by providing additional bits for

coding said region (section 0023, in this segment, van der Schaar et al discusses

transmitting "designated areas" within an image at a higher priority than other areas of

the image. In other words, designated areas are regions of interest being coded with at a higher priority i.e. with more bit allocation than other areas of the image).

Re claim 3, the method as defined by claim 1, wherein said step of providing additional bits for coding said region comprises providing additional bits for said region in the compressed enhancement layer bit stream (w/r to discussion in claim 1, also see sections 0023-0024).

Re claim 5, the method as defined by claim 3, wherein said processing to produce a compressed enhancement layer bit stream includes a bit plane shifting step, and wherein said step of providing additional bits for said region includes increasing the bit shifting values in said region (sections 0025-0026).

Re claim 6, the method as defined by claim 1, wherein said step of processing said input video to produce a compressed base layer bit stream includes forming motion vectors, and wherein said step of identifying a region of interest in a video frame includes basing said identifying on said motion vectors (fig. 3a: "motion estimation/compensation", also sections 0037-0038, in these segments, the position of areas of interest are inherently provided by the motion estimation, which is used to detect motion vectors. It is noted that '161 is for MPEG-4, which is the coding protocol for image object segmentation or region-of-interest).

Re claim 7, the method as defined by claim 3, wherein said step of processing said input video to produce a compressed base layer bit stream includes forming motion vectors, and wherein said step of identifying a region of interest in a video frame

includes basing said identifying on said motion vectors (the limitations have analyzed and rejected w/r to claim 6 above).

Re claim 9, the method as defined by claim 6, wherein said step of identifying a region of interest in a video frame based on said motion vectors includes basing said identification on the magnitude of motion vectors (w/r to claim 6, also it is inherent that motion estimation disclosed in '161 not only detects motion vectors, but also provide identification on the magnitude of motion vectors because motion estimation needs to know the magnitude of motion vectors to identify the optimum motion vector for compensation).

Re claim 10, the method as defined by claim 6, wherein said step of identifying a region of interest in a video frame based on said motion vectors includes basing said identification on the intensity change of neighboring regions based on motion vectors (w/r to claims 6 and 9, also it inherent that motion estimation disclosed in '161 evaluate motion vectors based on identification on the intensity change of neighboring regions because this step has to take place in order to determine the minimum motion vector).

Re claim 11, the method as defined by claim 3, wherein said step of processing said input video to produce a compressed base layer bit stream includes forming motion vectors and determining motion compensation values, and wherein said step of identifying a region of interest in a video frame includes basing said identifying on said motion vectors and said motion compensation values (the limitations have been analyzed w/r to claims 6, 9 and 10, furthermore fig. 3a shows motion estimation and compensation are involved).

Claim Rejections - 35 USC § 103

3.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

4.   Claims 2, 4, 8 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable

over van der Schaar et al, US 2002/0006161 (hereinafter '161) as applied to claim 1

above and further in view of van der Schaar et al, US 6,501,797 (hereinafter '797).

Re claim 2, van der Schaar et al '161 discloses providing additional bits for

coding designated areas or region of interest for enhancement, but fails to disclose

whether said region comprises providing additional bits for said region in the

compressed base layer bit stream as claimed.

van der Schaar et al '797 discloses providing additional bits for said region in the

compressed base layer bit stream (fig. 3: 322, col. 7, line 39 to col. 8, line 19, in this

segment, van der Schaar et al discusses the output signals of base layer adjust or

monitor the operation of the enhancement rate allocator circuit "358").

Taking the combined teaching of van der Schaar et al '161 and '797 as a whole,

it would have been obvious to implement providing additional bits for coding designated

areas or region of interest for enhancement layer coding by providing additional bits for

said region in the compressed base layer bit stream as claimed for the benefit of

improving coding efficiency of the enhancement layer coding and improving image

quality ('797 col. 2, line 60 to col. 3, line 11).

Re claim 4, the method as defined by claim 2, wherein said processing to produce a compressed base layer bit stream includes a quantization step, and wherein said step of providing additional bits for said region includes decreasing the quantization step in said region (w/r to discussion in claim 2, furthermore, base layer allocator "322" regulates the bit amount by controlling the quantization "316". Decreasing quantization step translates to increase bit amount and vice versa).

Re claim 8, the method as defined by claim 4, wherein said step of processing said input video to produce a compressed base layer bit stream includes forming motion vectors, and wherein said step of identifying a region of interest in a video frame includes basing said identifying on said motion vectors (the limitations have analyzed and rejected w/r to claim 6 above. Claim 2 provides reasons/motivation for combined teaching).
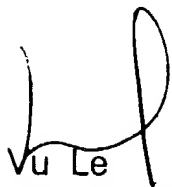
Re claim 12, the method as defined by claim 4, wherein said step of processing said input video to produce a compressed base layer bit stream includes forming motion vectors and determining motion compensation values, and wherein said step of identifying a region of interest in a video frame includes basing said identifying on said motion vectors and said motion compensation values (the limitations have been analyzed and rejected w/r to claim 11. Claim 2 provides reasons/motivation for combined teaching).

Contact

5.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Vu Le whose telephone number is 703-308-6613. The

examiner can normally be reached on M-F 8:30-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Chris Kelley can be reached on 703-305-4856. The fax phone number for

the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

Vu Le

Primary Examiner

AU 2613

(703) 308-6613

Vu.Le@uspto.gov

		Application/Control No.	Applicant(s)/Patent Under Reexamination
Notice of References Cited		09/977,081	LI, WEIPING
		Examiner	Art Unit	
		Vu Le	2613	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-2003/0165274 A1	09-2003	Haskell et al.	382/243
	B	US-2002/0009141	01-2002	Yamaguchi et al.	375/240.11
	C	US-6,263,022 B1	07-2001	Chen et al.	375/240.03
	D	US-2002/0181580 A1	12-2002	Van Der Schaar, Mihaela	375/240.01
	E	US-5,508,746 A	04-1996	Lim, Jae S.	375/240.25
	F	US-6,501,797 B1	12-2002	van der Schaar et al.	375/240.12
	G	US-2002/0006161	01-2002	Van Der Schaar et al.	375/240.11
	H	US-6,519,285 B2	02-2003	Yamaguchi et al.	375/240.12
	I	US-6,707,949 B2	03-2004	Haskell et al.	382/243
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Illgner et al, "Spatially Scalable Video Compression Employing Resolution Pyramids", IEEE Journal On Selected Areas In Communications, Vol. 15, No. 9, pp. 1688-1703, Dec. 1997.
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

U.S. Patent and Trademark Office
PTO-892 (Rev. 01-2001)   **Notice of References Cited**   Part of Paper No. 4

Spatially Scalable Video Compression Employing Resolution Pyramids

Klaus Illgner and Frank Müller

Abstract— In this paper, a spatially scalable video coding scheme for low bit rates is proposed. The codec is especially well suited for communications applications because it is based on motion-compensated predictive coding which provides the necessary low-delay property. The frames to be coded are decomposed into a Gaussian pyramid. Motion estimation and compensation are performed between corresponding pyramid levels of successive frames. We show that, to fulfill specific needs of spatial scalability, the motion compensation on each level must result in compatible prediction errors (displaced frame differences, DFD). Compatibility of the prediction errors means that the pyramid formed by independently obtained DFD's (the DFD pyramid) is close to a Gaussian pyramid decomposition of the DFD of the highest resolution level. From the DFD pyramid, a least squares Laplacian pyramid is derived, which is quantized and coded. The DFD encoder outputs an embedded bit stream. Thus, the coder control may truncate the bit stream at any point, and can keep a fixed rate. The motion vector fields obtained at the different resolution levels are also encoded by employing a pyramid approach. Simulation results show that the proposed coder achieves a coding gain compared to simulcast coding.

Index Terms—Laplacian pyramid, multiresolution pyramid, scalable motion compensation, spline filter, video compression.

I. INTRODUCTION

THE rapid development of networks leads to increasing interest in video communications. All current standards for communications-related video compression (H.261, H.263) are based on a single resolution hybrid (i.e., motion-compensated predictive) coding scheme. However, bandwidth limitations, multipoint operation with receivers of different capabilities, and bandwidth-dependent charging are good reasons to use scalable coding schemes. Also, MPEG-4 announced strong demands for scalable video coding algorithms.

The term *scalability* is ambiguous because it is used in a spatial, temporal, or SNR context. In this paper, the focus is on spatial multiresolution schemes since this is an essential prerequisite to provide the property of *spatial* scalability. This means that the receiver can reconstruct frames of smaller spatial resolution using only a subset of the complete bit stream, thus fulfilling the above-mentioned demands.

In search of an appropriate coding principle, spatiotemporal subband or wavelet approaches are problematic because the coding delay must be kept as small as possible for communications applications. A promising approach is therefore to extend the classical hybrid coding principle, which is the basis of

most coding schemes for communications, to fulfill the needs of spatial scalability.

Known multiresolution approaches based on the hybrid coding principle utilize several DPCM loops on different resolution levels. The design objective of a spatially scalable video codec is to find optimal predictions for each resolution level. However, this task cannot be solved straightforwardly because the prediction errors have to be coded efficiently. This can be accomplished only by exploiting the dependencies between the different resolution levels. The degree of coupling between the DPCM loops makes for the main difference between various schemes.

A straightforward approach is to decompose the frames into multiple resolution frames, and to code each resolution level independently. However, this approach results in significant coding overhead. The scalable codec described in [1] is based on MPEG-II, and employs several DPCM loops on different resolution levels. There exists a slight interconnection between the layers since motion estimation is performed in a hierarchical fashion.

A tighter interconnection is realized in [2], where a two-layer scalable pyramid codec based on two DPCM loops is described. Hierarchical motion estimation is employed in combination with a hierarchical VQ on the displaced frame differences. There exist also various two-dimensional (2-D) wavelet-based and subband schemes [3]–[5]. Motion estimation and compensation turn out to be difficult in the wavelet/subband domain. One difficulty arises from the shift variance of downsampling in the wavelet domain [3]. Another reason is decreased efficiency of motion compensation on bandpass frames since the high-pass signal components are disturbed due to noise and aliasing. Therefore, most approaches perform motion compensation on low-pass subbands.

In this paper, a hybrid spatially scalable video coding algorithm is presented which is based on a Gaussian pyramid decomposition. Motion estimation and compensation are performed between each pyramid level such that the resulting prediction errors are close to a Gaussian pyramid decomposition of the prediction error at the highest resolution level. For coding of the displaced frame differences (DFD) as well as the motion vector fields, the statistical dependencies between the pyramid levels (corresponding to the different resolutions) are utilized by coding jointly all resolutions with a compact Laplacian pyramid representation. More specifically, the DFD is decomposed into a centered Laplacian least squares spline pyramid (construction and properties of this kind of pyramid are derived in this paper), and subsequently coded in an embedded fashion.

One of the most efficient embedded image coding algorithms has been developed by Shapiro [6], and is called the embedded zero-tree wavelet (EZW) coding algorithm. It consists of a transformation of the image into a wavelet domain, and subsequent embedded coding with the zero-tree algorithm, which starts with coarse quantizations and refines the wavelet coefficients in subsequent passes. Taubman [7] developed a three-dimensional (3-D) subband video coder, which (restricted to 2-D) resembles the EZW coder in the aspects of decomposition and layered quantization. Instead of using zero-trees for coding of significance maps, he used a conditioning context to predict zeros in dominant passes. He shows that this kind of general prediction context provides significant gain over the zero-tree coding method. In both cases, a wavelet decomposition has been used. In a previous work [8], we adapted the EZW coding scheme to a Laplacian pyramid decomposition, and discussed the usage of conditional arithmetic coding in a Laplacian pyramid. We showed there that the conditional arithmetic coder outperforms the pyramid zero-tree coder.

In this paper, we adapt the Laplacian pyramid coding of displaced frame differences to a spatially scalable coding scheme. We have to deal with two DFD's per frame, corresponding to the base layer and the refinement layer of the codec. The DPCM loops for both layers are coupled such that the DFD in the base layer is "close" to a reduced version of the DFD of the refinement layer. This requires a careful design of the filter operators which accomplish the transition between the different resolutions. Requirements for these filters are given in this paper, together with a new filter design which fulfills these requirements.

The paper is organized as follows. In the next section, an overview of the hybrid spatially scalable coding concept is given. The third section discusses motion compensation for scalable video coding in general, and mentions particular implementation issues of the proposed scheme. In the fourth section, desirable properties for the reduce and expand operators (which define the pyramid decompositions) are summarized. While most of these properties can be fulfilled with any least squares pyramid [9], employment in the scalable codec makes the *centering* a desired feature. The derivation of such pyramids is included in the Appendix. The fifth and sixth sections describe DFD coding and vector field coding in detail, and the last two sections are devoted to simulation results and some conclusions.

II. DESIGN OF A SPATIALLY SCALABLE CODING SCHEME

We start with a discussion of the motion-compensated predictive coder shown in Fig. 1. The current frame is denoted by g_n and the prediction of g_n by $\hat{g}_n$. The DFD $d_n = g_n - \hat{g}_n$ is decomposed into a Gaussian pyramid. A Laplacian pyramid is derived from the Gaussian pyramid and coded in embedded fashion using arithmetic coding. The prediction $\hat{g}_n$ is calculated as a motion-compensated version of the previously reconstructed frame $\tilde{g}_{n-1}$ using overlapping blocks. For motion estimation between the frames g_n and $\tilde{g}_{n-1}$, a gain–cost criterion is employed. The motion vector field v_n is
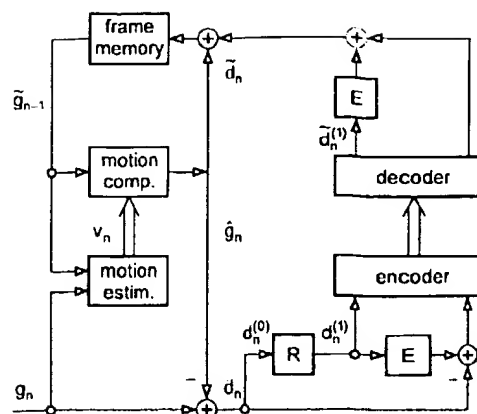

Fig. 1. Block diagram of *PYRACO*, a predictive multiresolution video coding scheme (nonscalable version).
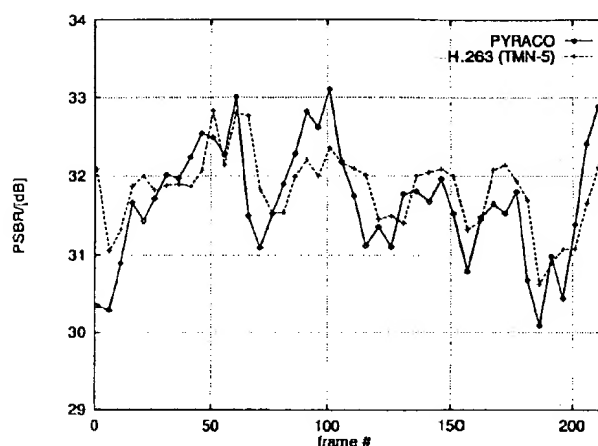

Fig. 2. PSNR of sequence *foreman* coded at 48 kbit/s and 5 fps (fixed).

coded losslessly in a hierarchical fashion [10]. Fig. 2 shows[1] that this coding approach, which is termed PYRACO, reaches approximately the same coding performance as the currently best reference coder H.263 (TMN-5 implementation) [11]. An important difference is that PYRACO outputs a constant bit rate per frame, which simplifies coder control and reduces coding delay, while TMN generates a significantly varying rate.

The multiresolution image representation as a Gaussian pyramid offers a natural design choice for a spatially scalable coding scheme with multiple resolution levels. For simplicity of presentation, we consider in the following only two resolution levels. A block diagram of the spatially scalable hybrid encoder is depicted in Fig. 3. The *base layer* coder emits the lower spatial resolution, and follows the same coding principle as the coder in Fig. 1. A temporally predictive coding approach is also used for encoding of the *refinement layer*, which provides the higher spatial resolution.

[1] The test sequence has been obtained from the CCIR sequence using spline filters [9] for preserving sharpness. The TMN codec runs with all options except PB frames.
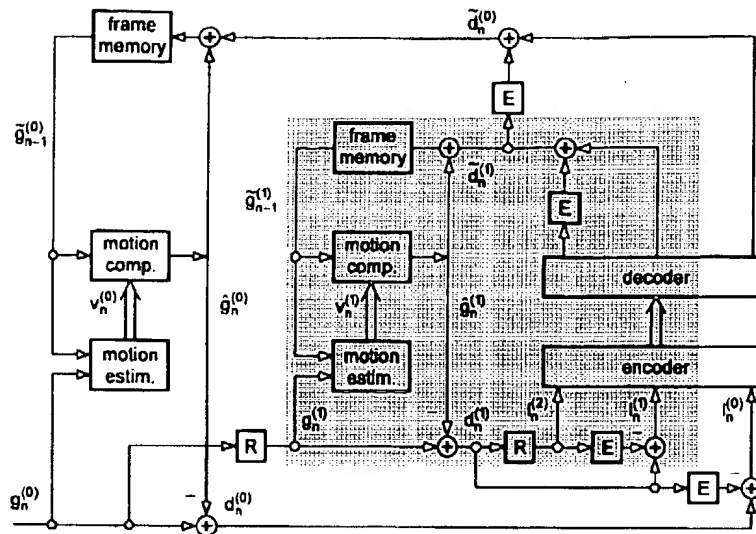
Fig. 3. Encoder of a two-level scalable predictive video coding scheme (the base layer is shaded).
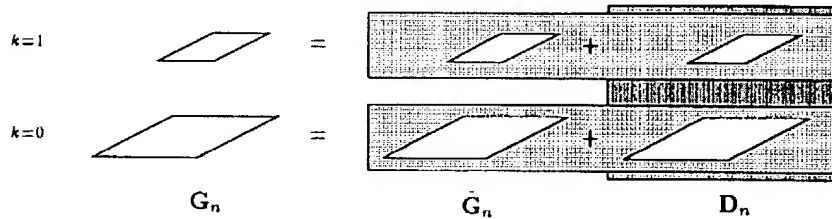

Fig. 4. Relations between two resolutions in a predictive video coding scheme.

A. Aim for Motion Compensation to Achieve Scalability

The first step is a decomposition of the current frame g_n into a Gaussian pyramid G_n of two levels[2]

$$G_n = (g_n^{(0)}, g_n^{(1)}) = (g_n, R(g_n)). \qquad (1)$$

$R(\cdot)$ denotes the so-called reduce operator, which consists of a linear low-pass filter followed by subsampling of the image in each dimension by a factor of 2. The aim is now to employ a predictive video compression scheme for both resolution levels of the frame to be coded. Hence, both levels are decomposed into a predicted frame and the prediction error, which is depicted in Fig. 4 reading the figure horizontally. However, reading Fig. 4 vertically, the prediction errors of both resolution levels form a pyramid. This pyramid can be coded efficiently [8] if it is a Gaussian-type pyramid.

Hence, the idea for the design of a spatially scalable predictive video compression scheme is to calculate predictions $\hat{g}_n^{(k)}$ for each level k

$$d_n^{(k)} = g_n^{(k)} - \hat{g}_n^{(k)}, \qquad k = 0, 1$$

such that the relation

$$R\left(d_n^{(0)}\right) \overset{!}{=} g_n^{(1)} - \hat{g}_n^{(1)} \qquad (2)$$

[2] Throughout the paper, we use the convention that level 0 corresponds to the refinement layer, and level 1 to the base layer.

holds, which is equivalent to the condition

$$R\left(\hat{g}_n^{(0)}\right) \overset{!}{=} \hat{g}_n^{(1)}. \qquad (3)$$

The predicted levels are obtained by motion compensation of the corresponding level of the previous frame:

$$\hat{g}_n^{(k)} = \mathrm{MC}\left(\tilde{g}_{n-1}^{(k)}, v_n^{(k)}\right).$$

If the predictions satisfy (3), the prediction errors in Fig. 4 indeed constitute a Gaussian pyramid D_n.

In a spatially scalable coding scheme, the prediction for the base layer needs to be calculated *without* considering the higher resolution level. This constraint is the central difficulty in the design.

Since the operator $\mathrm{MC}(\cdot)$ is nonlinear and space variant, motion compensation and low-pass filtering cannot be interchanged in general. Furthermore, subsampling is a shift-variant operation causing additional aliasing errors. Therefore, the low-pass filtered prediction from level $k = 0$ differs from the prediction on level $k = 1$, even if the motion is known exactly at level 0 [13]. Hence, equality in (3) cannot be obtained in general.

In order to achieve high compression efficiency, the aim is to find predictors for each scalable level such that (3) is approximated sufficiently accurate

$$R\left(\hat{g}_n^{(0)}\right) \approx \hat{g}_n^{(1)}. \qquad (4)$$

The design objective is to make the prediction of the base layer similar to the *reduced* prediction of the higher resolution level. The approach followed in this paper calculates the base layer prediction on an expanded frame, and subsequently applies a reduce operator. It is shown in Section III that by appropriately selecting the expand and reduce filters, (4) can be achieved sufficiently accurate. It turns out that the *same* filters as for the DFD pyramid coding are suitable.

B. Pyramid Decomposition of the Base Layer DFD

The base layer DFD $d^{(1)}$ is decomposed into a Laplacian pyramid using centered cubic splines (see Appendix B), which results in high energy concentration into the higher pyramid levels. The decomposition follows the *Gaussian–Laplacian* pyramid approach [12]. First, an intermediate pyramid structure, the Gaussian DFD pyramid D', is generated. The pyramid is initialized at the lowest level $l = 1$ with the original base layer DFD we want to code. From this level, a coarse approximation is derived by applying a reduce operator $R(\cdot)$. This procedure is iterated, resulting in the base layer Gaussian pyramid

$$D' = \left\{ \left(d^{(1)}, \cdots, d^{(L-1)} \right) | d^{(k)} = R\left(d^{(k-1)} \right), \right.$$
$$\left. k = 2, \cdots, L - 1 \right\} \qquad (5)$$

consisting of $L - 1$ levels.[3]

For computation of the corresponding Laplacian pyramid, an expand operator $E(\cdot)$ is defined, which predicts a level of the Gaussian pyramid from the next higher level, thus expanding the size again by a factor of 2 for each dimension. The expand operator consists of upsampling and subsequent filtering with a linear interpolation filter. The Laplacian pyramid L' captures the loss of information occurring through the reduction of the spatial resolution by means of the difference between a Gaussian image at level k and the expanded version of the Gaussian image at level $k + 1$:

$$L' = \left(l^{(1)}, l^{(2)}, \cdots, l^{(L-1)} \right) \qquad (6)$$

where

$$l^{(k)} = d^{(k)} - E\left(d^{(k+1)} \right), \qquad k = 1, \cdots, L - 2.$$

The top level of L' is "initialized" with the top level of D'

$$l^{(L-1)} = d^{(L-1)}.$$

Knowledge of the Laplacian pyramid is sufficient to reconstruct the base layer DFD $d^{(1)}$. Thus, an encoder can build up a coarse representation and successive refinements (the Laplacian levels) of an image, and a decoder can reconstruct the original from this information.

---

[3] According to our convention, the complete pyramid consists of L levels, and therefore the base layer pyramid has $L - 1$ levels.

C. Approach for DFD and Vector Field Coding

As is depicted in the block diagram of the spatially scalable hybrid encoder in Fig. 3, motion estimation and compensation are performed on the resolution levels 0 and 1, yielding a prediction error for the base and the refinement layer. The base layer prediction error is decomposed further into a Gaussian pyramid according to (5). Because the receiver needs only to know about the expand operator, the levels of the pyramid can be decomposed by employing different reduce operators, which may even be nonlinear. Therefore, the complete Gaussian pyramid of the prediction errors D is obtained by concatenating the prediction error of the refinement layer and the Gaussian pyramid of the base layer

$$D = (d^{(0)} \ D'). \qquad (7)$$

From this pyramid, the *complete* Laplacian pyramid

$$L = \left\{ \left(l^{(0)}, l^{(1)}, \cdots, l^{(L-1)} \right) | l^{(k)} = d^{(k)} - E(d^{(k+1)}), \right.$$
$$\left. k = 0, \cdots, L - 2, \ l^{(L-1)} = d^{(L-1)} \right\} \qquad (8)$$

is calculated, which is sufficient for reconstruction of the DFD at the refinement layer decoder. The base layer decoder needs only to decode the truncated Laplacian pyramid L'. The Laplacian pyramid is quantized with a layered quantization scheme. The coefficients are selected and quantized according to their amplitude in decreasing order. Due to the energy concentration, mainly coefficients from higher levels are coded first. The resulting symbol streams are coded using conditioning contexts and adaptive arithmetic coding [8]. Since the context choice depends only on the current or higher levels, scalability is retained. A detailed description of the coding scheme and the filters is given in Section V.

The open-loop approach, where the Laplacian pyramid levels are quantized independently, simplifies layered quantization, and allows for embedded coding. Furthermore, as is shown in (41) in Appendix A, the feedback of the quantization errors of the Laplacian pyramid within the DPCM loop does not degrade the performance since the current level $l_n^{(k)}$ is affected only by the quantization noise of the corresponding level of the previous frame $l_{n-1}^{(k)}$.

The Laplacian pyramid coding principle is adopted for coding of the motion vector fields. The main differences are that for both vector field components independent Laplacian pyramids are calculated, and the pyramids are coded losslessly. The vector field of the base layer is decomposed into Laplacian pyramids, and to obtain the complete pyramid, the motion vector differences of the refinement layer are added as level 0. Since the vector field components have finite precision, quantization is replaced by bit plane coding. This topic is elaborated in Section VI.

III. MOTION ESTIMATION AND COMPENSATION

The motion model used for this coder concept assumes that the motion of the 3-D objects can be described completely by displacements $v_n(x)$ on the image plane. Denoting the location of the pixels on the image grid by $x = (x, y)^T$, frame g_n is related to g_{n-1} by

$$g_n(x + v_n(x)) = g_{n-1}(x)$$

neglecting occlusions, uncovered background, and global illumination changes. The displacement vector field for frame g_n is the set of all displacement vectors v_n.

The aim of motion estimation in the context of coding is to find displacements $\hat{v}_n(x)$ with respect to the previously decoded frame $\tilde{g}_{n-1}$

$$\hat{v}_n = \text{ME}(g_n, \tilde{g}_{n-1})$$

such that the prediction error

$$e_{\text{DFD}} = \sum_x \|g_n(x) - \hat{g}_n(x)\| \tag{9}$$

becomes minimal according to a norm. The prediction signal $\hat{g}_n$ is obtained typically by displacing pixels of the previous frame, which may be weighted additionally (e.g., overlap block motion compensation) using a weighting function w_i

$$\hat{g}_n(x) = \sum_i w_i(x) \cdot \tilde{g}_{n-1}(x - \hat{v}_n(x_i)) = \text{MC}(\tilde{g}_{n-1}, \hat{v}_n). \tag{10}$$

A. Optimal Motion Compensation for the Base Layer

For the moment, we assume that the original previous frame is used for "motion compensation." On the base layer, motion compensation is performed on low-pass filtered and downsampled frames, which limits the prediction performance even if the motion is known. One reason is that subsampling causes aliasing errors. Another one is that motion compensation and low-pass filtering cannot be interchanged in general

$$f_R\left(\text{MC}\left(g_{n-1}^{(k)}, v^{(k)}\right)\right) \approx \text{MC}\left(f_R\left(g_{n-1}^{(k)}\right), v^{(k)}\right) \tag{11}$$

because the low-pass filter $f_R(\cdot)$ weights and sums up spatially neighbored pixels. Thus, the filtered pixels are different if the original pixels are displaced prior to filtering [13]. An exception is the case of constant displacements $v(x) = \text{const.}$, where all pixels undergo the same displacement. Low-pass filtering and motion compensation are then interchangeable, and equality holds in (11).

A deeper analysis reveals that motion can be interpreted as a shearing of the image spectrum into a hyperplane along the ω_t axis in the 3-D frequency space, and motion compensation reverts the shearing [14]. Therefore, low-pass filtering prior to motion compensation causes degradations since the hyperplane rather than the plane of the original image spectrum is multiplied by the filter transfer function. This interpretation also reveals why motion compensation prior to filtering results in more accurate predictions than compensation after filtering.

According to our motion model, where the moving scene content results in displacements of pixels in the image plane of the camera, the next frame is generated by displacing the pixels of the previous frame followed by filtering and subsampling. Therefore, for motion compensation of the base layer using a given motion vector field $v_n^{(0)}$, the low-pass filtered prediction from the highest available resolution level results in a higher prediction gain than calculating the prediction on the lower resolution level [13]

$$\hat{g}_{n,\text{opt}}^{(k)} = R^k\left(\text{MC}\left(g_{n-1}^{(0)}, v_n^{(0)}\right)\right), \qquad k = 0, 1. \tag{12}$$

This type of prediction is termed in the following *optimum prediction*. The superscript k denotes that the reduce function is applied k times. In principle, (12) holds also if the *reconstructed* previous frame is used for motion compensation. Hence, the Gaussian-type pyramid D_n formed by the displaced frame differences

$$d_n^{(k)} = g_n^{(k)} - \hat{g}_{n,\text{opt}}^{(k)}, \qquad k = 1 \cdots L - 1$$

has minimal energy. Furthermore, due to the linearity of the reduce operator

$$\begin{aligned} R(d_n^{(k)}) &= R\left(g_n^{(k)} - \hat{g}_{n,\text{opt}}^{(k)}\right) \\ &= R\left(g_n^{(k)}\right) - R\left(\hat{g}_{n,\text{opt}}^{(k)}\right) \\ &= g_n^{(k+1)} - \hat{g}_{n,\text{opt}}^{(k+1)} \\ &= d_n^{(k+1)} \end{aligned}$$

holds for each level of D_n. Thus, D_n is identical to the pyramid obtained by decomposing the prediction error of level $k = 0$. It is shown in [8] that such a pyramid can be coded efficiently.

B. Scalable Motion Compensation

This is the crucial part in a spatially scalable coding scheme since the prediction for the base layer needs to be calculated without considering the refinement level. The design aim with respect to (2) is to achieve equality of

$$R\left(\text{MC}\left(g_{n-1}^{(0)}, v_n^{(0)}\right)\right) \approx \text{MC}\left(R\left(g_{n-1}^{(0)}\right), v_n^{(1)}\right) \tag{13}$$

with $v_n^{(1)}$ being a suitably reduced version of $v_n^{(0)}$. Due to (11), an optimal prediction for the base layer $g_n^{(1)}$ cannot be obtained from $g_{n-1}^{(1)}$ in general. Therefore, the aim is to approximate $\hat{g}_{n,\text{opt}}^{(1)}$ as closely as possible which, with respect to (12), is equivalent to approximating the refinement layer $g_{n-1}^{(0)}$ as accurately as possible. The idea is to interpolate the base layer frame $g_{n-1}^{(1)}$ such that the distance to the refinement layer $g_{n-1}^{(0)}$ is minimized in the least squares sense:

$$\left|g_{n-1}^{(0)} - E\left(g_{n-1}^{(1)}\right)\right|^2 \longrightarrow \min. \tag{14}$$

In real coders, only the *coded* previous frame $\tilde{g}_{n-1}^{(k)} = g_{n-1}^{(k)} + q'^{(k)}_{n-1}$ is available, where q' denotes the quantization error (Appendix A). However, (14) remains valid since the quantization error of level 0 is fixed

$$\tilde{g}_{n-1}^{(0)} - E\left(\tilde{g}_{n-1}^{(1)}\right) = g_{n-1}^{(0)} - E\left(g_{n-1}^{(1)}\right) + q_{n-1}^{(0)}.$$

Motion compensation is performed on the expanded frames, and the final prediction is obtained by reducing the compensated, expanded frame

$$\hat{g}_n^{(k)} = R\left(\text{MC}\left(E\left(\hat{g}_{n-1}^{(k)}\right), v_n^{(k)}\right)\right), \qquad k = 0, 1. \tag{15}$$

Since the equation can be interpreted for the base layer as half-pel motion compensation (assuming interger valued $v_n^{(0)}$), the same technique is also applicable for the refinement layer

$k = 0$. The vector fields $v_n^{(k)}$ are obtained by motion estimation on the expanded frames

$$v_n^{(k)} = \text{ME}\left(E\left(g_n^{(k)}\right), E\left(\tilde{g}_{n-1}^{(k)}\right)\right), \qquad k = 0, 1. \qquad (16)$$

Note that (15) depends on no particular method for motion estimation and compensation.

To achieve a good approximation of $\hat{g}_{n,\text{opt}}^{(1)}$, the operator pair $E(\cdot), R(\cdot)$ needs to meet further constraints. The chain of operators must equal the identity operator

$$R\left(E\left(g_{n-1}^{(k)}\right)\right) = g_{n-1}^{(k)}.$$

This characteristic ensures that filtering effects do not distort the prediction. Furthermore, for the special case of global constant translational motion, the optimum prediction can be obtained.

Additionally, the filters must account for the fact that motion is a local property. Hence, the filters should be spatially localized or, in other words, the impulse response of the filters should have compact support.

The required properties are provided by centered third-order spline filters, a modified version of the cubic spline filters described in [15]. The class of spline filters allows for a tradeoff between spatial localization and aliasing since they include the Haar filter as a filter of order 0 as well as the sinc interpolator for infinite order. The distinctive feature of the proposed filters is the centering of the lower resolution grid with respect to the higher resolution grid. The advantage for scalable motion compensation is that the centered spline filters do not introduce a half-pel shift, and displacements remain at the same location in different levels. In Section IV, the filter design is linked to the Laplacian pyramids, and a detailed description is given in Appendix B.

C. Implementation Aspects

Since in a scalable video coding scheme the refinement level is not available to the base layer encoder, motion estimation is performed top down starting at the base layer. The method for motion estimation and compensation can be chosen freely in principle. However, an approach based on block matching is simple and robust. Another advantage of a block-oriented scheme is that low-pass filtering and motion compensation are interchangeable for $v_n^{(k)} = \text{const.}$ (11). Therefore, locally using a translational motion model is advisable.

To obtain a smooth vector field, a gain–cost criterion is evaluated as distance measure for each block j [16]

$$v_n(j) = \underset{v \in \mathcal{V}}{\text{argmin}}\left\{\log\left(e_j^2(v)\right) + \lambda \cdot \sum_{i \in \mathcal{N}_j} \|v - v_n(i)\|\right\}$$

$$\lambda = \text{const.} \qquad (17)$$

with $\mathcal{N}_j$ denoting the set of neighbored blocks of block j. The calculation of the prediction error e_j for block j according to (9) employs overlapping blocks; hence, (17) includes overlap block motion estimation. Furthermore, motion compensation causes no blocking artifacts. The factor λ controls the smoothness by penalizing large vector differences. Due to the interaction with neighbored blocks, the vector field

needs to be calculated iteratively. However, the computational load is still less than for full search block matching since the test vector set $\mathcal{V}$ is very small (18 candidates), and only about four iterations are sufficient. Since estimating motion at the expanded frames (16) can be interpreted as half-pel motion estimation, no additional half-pel motion estimation procedure like bilinear interpolation is needed.

At the refinement layer, the same gain–cost constrained block matching scheme applies. The vector field $v_n^{(1)}$ from the base layer is taken into consideration. Due to the constraint of (2), the predictions must be consistent across scales. Hence, the size of the blocks is increased by a factor of 2. A block size of 16×16 at level 0 corresponds to a block size of $16 \cdot 2^{-k} \times 16 \cdot 2^{-k}$ at level k. As a consequence, the vector field sizes of the scalable resolution levels are *equal*. The vector field $\hat{E}_v(v_n^{(1)})$ serves as an initial estimate for motion estimation, calculated by an appropriate scaling function $\hat{E}_v$. Since the number of vectors at both levels is the same, $\hat{E}_v$ scales only the amplitudes of the vector components by 2. Regarding $v_n^{(k)}$ as average motion, the motion vector field $v_n^{(k-1)}$ can be written as

$$v_n^{(k-1)} = \hat{E}_v\left(v_n^{(k)}\right) + \Delta v_n^{(k-1)} \qquad (18)$$

where $\Delta v_n^{(k-1)}$ denotes the refinement. The gain–cost criterion constrains the motion estimate to smooth vector fields eliminating spurious vectors. Finally, a hierarchy of motion vector fields has been obtained, and needs to be coded (Section VI).

IV. CHOICE OF REDUCE AND EXPAND OPERATORS

The choice of the reduce operator $R(\cdot)$ and the expand operator $E(\cdot)$ is the key issue to achieve efficient multiresolution motion estimation and compensation, as well as efficient coding of the Laplacian pyramid of the displaced frame differences.

In the previous section, we compiled operator conditions for multiresolution motion compensation. First, the interpolation of the base layer should approximate the refinement layer in the least squares sense. Furthermore, the operator chain $E(\cdot)$ followed by $R(\cdot)$ should be the identity operator.

For DFD coding, the main objective for the filter design with respect to the Laplacian pyramid is to shift the signal energy into the higher pyramid levels. Concentration of the energy in only few pixels results in the desired coding gain. This is equivalent to the objective of minimizing the energy of the lower levels of the Laplacian pyramid. A useful criterion is that the approximation operator $[R(\cdot)$ followed by $E(\cdot)]$ should be optimal in the least squares sense.

Comparing both sets of conditions, it turns out that they are identical for both tasks. Hence, one set of filters suffices. We chose spline filters, and found that centered third-order filters performed best. In Appendix B, the filters are described in detail, and it is proven that they provide the desired properties.

V. DFD PYRAMID CODING

The two-layer scalable coding scheme under consideration outputs two displaced frame differences $d^{(0)}, d^{(1)}$ for each coded frame. The base layer DFD $d^{(1)}$ is decomposed into

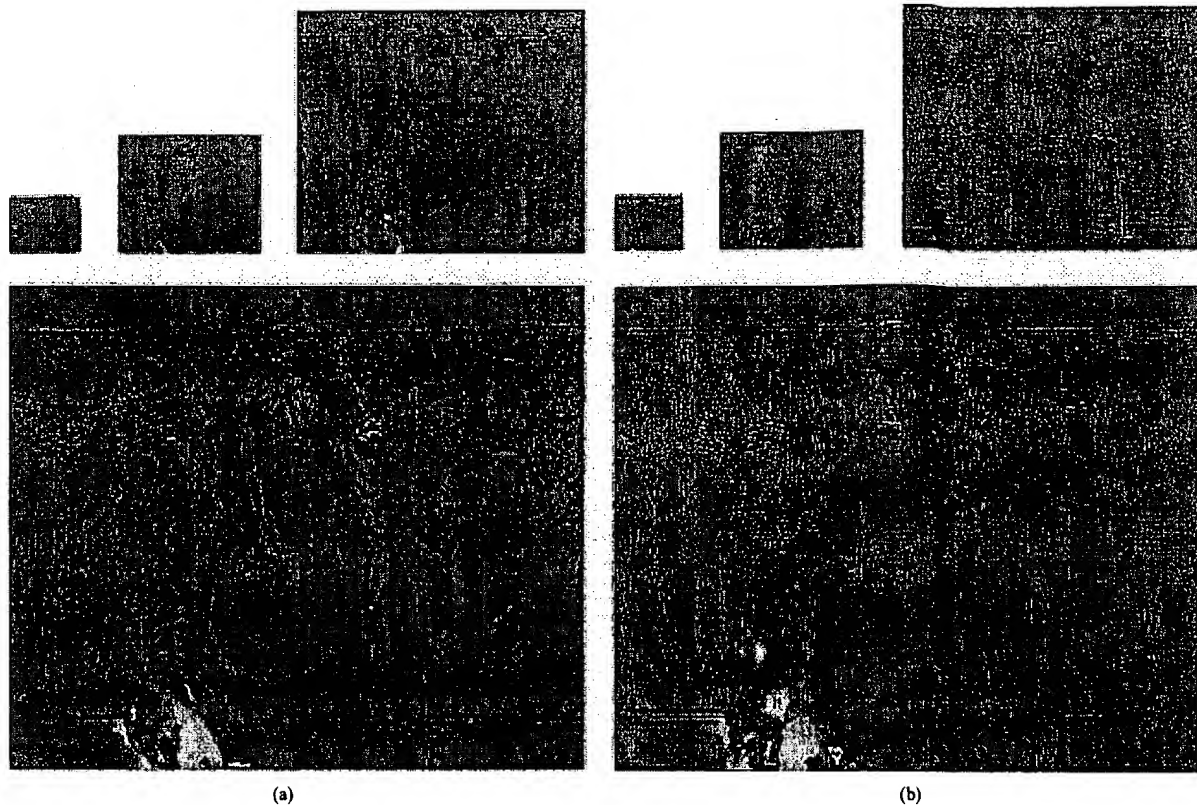(a)   (b)

Fig. 5.   Least squares Laplacian pyramid (LSLP) decomposition of DFD 11 of sequence *Silent*. For display, the amplitudes have been multiplied by two for the Gaussian and by four for the Laplacian pyramid. (a) Gaussian DFD pyramid D. (b) Laplacian DFD pyramid L.

a Laplacian pyramid. As explained in Section II, a Laplacian level $l^{(0)}$ is derived from the refinement layer DFD $d^{(0)}$, and this level builds together with the truncated pyramid L' the complete Laplacian pyramid of both DFD's.

This results in energy concentration into the higher (low-pass) levels of the pyramid, which consist of considerably fewer pixels than the original image. Usually, the energy is not equally distributed within the levels. This is due to the nature of motion compensation: some areas in the original sequence may contain complex motion where the motion compensation fails, while other areas contain no or only simple translational motion. The DFD energy will be the highest in areas where motion compensation failed. The severely limited bit budget for DFD coding in low bit-rate coders only allows encoding of selected areas of the DFD. Most DFD coding schemes encode the address information as overhead (addressing the areas, which are actually coded) separately from the quantized data. We use instead a layered coding method, where address information and data are encoded jointly in significance maps.

The coding gain which can be achieved with the Laplacian pyramid depends strongly on the employed filters. We have chosen a centered version of the cubic spline *least squares Laplacian pyramid* (LSLP) because this set of filters performed best of all investigated linear filters (see Appendix B).

In Fig. 5, the LSLP decomposition is shown for the frame difference between frame 11 and the motion-compensated prediction for this frame obtained from the reconstructed frame

6 of the image sequence "silent." The lowest level of the pyramid consists of 352×288 pixels (common interchange format, CIF). In Fig. 5(a), the Gaussian DFD pyramid D is given; in Fig. 5(b), the corresponding Laplacian pyramid L is given. The top three images in each subfigure constitute the truncated pyramids D' and L' (base layer DFD and Laplacian pyramid). The large images $d^{(0)}$ and $l^{(0)}$ belong to the refinement layers.

One can see that the base layer decomposition of $d^{(1)}$ into the truncated pyramid L' yields energy compaction into the higher pyramid levels. The design objective $R(d^{(0)}) \approx d^{(1)}$ does not hold with equality; therefore, $l^{(0)}$ contains more energy than the other levels of the Laplacian pyramid. Especially where strong motion appears (the raising hand), $d^{(0)}$ cannot be predicted well from $d^{(1)}$. On the other hand, $l^{(0)}$ contains much less energy than $d^{(0)}$, which indicates that predicting $d^{(0)}$ from $d^{(1)}$ is better than no prediction at all. Doing without prediction leads to the simulcast case (independent coding of base and refinement layer), which is used for comparison in Section VII.

A. Layered Quantization

Let $l^{(l)}(i, j)$ denote the pixel value at location (i, j) of level l of the pyramid L. The range of the indexes is $i \in 0 \cdots I/2^l - 1$, $j \in 0 \cdots J/2^l - 1$, and $l \in 0 \cdots L - 1$, where I and J are the dimensions of the original frame. The pyramid is scanned top down (starting with $l = L - 1$ down to $l = 0$), resulting in a

sequence of pyramid values $x[i]$. The scan order inside each level is a simple line scan.

Scalability is achieved by progressive quantization in a sequence of up to N layers. Each layer can be described by the associated quantizer. It is convenient to differentiate between two sets of quantizers $Q_0^D, \cdots, Q_{N-1}^D$ and $Q_1^S, \cdots, Q_{N-1}^S$. The first set of quantizers belongs to the *dominant passes*, and the second to the *subordinate passes* of the coding algorithm. These passes alternate, starting with a dominant pass. Hence, during the coding process, the sequence $Q_0^D, Q_1^S, Q_1^P, Q_2^S, Q_2^P, \cdots$ of quantizers is employed.

Each quantizer Q_n is defined by a set of disjoint quantization intervals $\mathcal{I}_{n,k_1}, \mathcal{I}_{n,k_2}, \cdots$ and the quantization function

$$Q_n^D(x) = k, \qquad \text{for } x \in \mathcal{I}_{n,k}^D \qquad (19)$$

$$Q_n^S(x) = k, \qquad \text{for } x \in \mathcal{I}_{n,k}^S \qquad (20)$$

which maps each input value x to an index k.

In order to allow efficient layered quantization, the sequence of quantization intervals must form a set of nested intervals. The "dominant" intervals $\mathcal{I}_{n,k}^D$ are symmetric around zero, and are uniformly spaced with a dead zone twice as large as all of the other quantization intervals. From layer to layer, each quantization interval is halved. Thus, the intervals are specified with a single parameter Δ_0 as follows:

$$\mathcal{I}_{n,k}^D = \begin{cases} (-\Delta_n, \Delta_n), & \text{if } k = 0 \\ [k\Delta_n, (k+1)\Delta_n), & \text{if } k > 0 \\ ((k-1)\Delta_n, k\Delta_n], & \text{if } k < 0 \end{cases} \qquad (21)$$

with

$$\Delta_n = \Delta_{n-1}/2, \qquad \text{for } n > 0. \qquad (22)$$

The "subordinate" intervals $\mathcal{I}_{n,k}^S$ are refinements of the intervals $\mathcal{I}_{n-1,k}^D$. More precisely, each interval $\mathcal{I}_{n-1,k}^D$ contains two intervals $\mathcal{I}_{n,k}^S$ of equal width, except for $k = 0$, where $\mathcal{I}_{n-1,0}^D = \mathcal{I}_{n,0}^S$. Hence, the "subordinate" intervals are symmetric around zero, uniformly spaced with a dead zone four times larger than the other intervals

$$\mathcal{I}_{n,k}^S = \begin{cases} \mathcal{I}_{n-1,0}^D, & \text{if } k = 0 \\ \emptyset, & \text{if } k \in \{-1,1\} \\ \mathcal{I}_{n,k}^D, & \text{if } k \notin \{-1,0,1\}. \end{cases} \qquad (23)$$

Note that each quantization interval Q_n^D is contained in some quantization interval of Q_{n-1}^D. Moreover, the intervals $\mathcal{I}_{n,0}^D$ contain the *three* intervals $\mathcal{I}_{n+1,-1}^D, \mathcal{I}_{n+1,0}^D, \mathcal{I}_{n+1,1}^D$, whereas all other intervals $\mathcal{I}_{n,k}^D$ (with $k \neq 0$) contain *two* intervals $\mathcal{I}_{n+1,2k}^D, \mathcal{I}_{n+1,2k+1}^D$. (Similar properties exist for the "subordinate" intervals since these are defined through (23) via the dominant intervals.) This is the only condition on the quantization intervals which must be imposed. However, specifying the intervals as in (21) is very convenient because the set of quantization functions is defined by Δ_0 and N alone.

B. Symbol Stream Generation

We define two sets of sequences $\sigma_n'[i]$ and $\delta_n'[i]$ as follows:

$$\sigma_n'[i] = Q_n^D(x[i]) \qquad (24)$$

$$\delta_n'[i] = Q_n^S(x[i]) \pmod 2. \qquad (25)$$

Since the quantization intervals form a set of nested intervals, all information necessary to recover the current quantization intervals is contained in these sequences. Moreover, if these sequences are ordered in the same way as the quantizers, i.e., $\sigma_0'[i], \delta_1'[i], \sigma_1'[i], \cdots$, a part of this sequence is still redundant. Particularly, 1) every $\delta_n'[i]$ for which $\sigma_{n-1}'[i] = 0$, and 2) every value $\sigma_n'[i] \notin \{-1,0,1\}$ can be predicted from previous parts of the alternating sequence $\sigma_0'[i], \delta_1'[i], \sigma_1'[i], \cdots$.

Thus, the variable length strings $\sigma_0, \delta_1, \sigma_1, \cdots$ contain the same information as the strings $\sigma_0'[i], \delta_1'[i], \sigma_1'[i], \cdots$, if we denote by σ_n and δ_n the strings obtained by removal of the redundant entries of $\sigma_n'[i]$ and $\delta_n'[i]$, respectively.

Removal of the redundant entries works the same way as in Shapiro's EZW coder [6]. In the first dominant pass, all entries are scanned; thus, σ_0' equals σ_0. The first subordinate pass deals only with the samples which have become significant in the first dominant pass. In accordance with condition 1), those values of $\delta_1'[i]$, for which $\sigma_0'[i]$ has been zero (insignificant), are omitted in δ_1. Condition 2) means that all samples which have become significant during previous dominant passes are not regarded in any subsequent dominant pass.

The symbol stream consisting of the concatenation of the strings $\sigma_0, \delta_1, \cdots$ is encoded arithmetically, which is described in the next subsection.

C. Conditional Arithmetic Coding

The strings δ_n are binary, and show only very little correlation between the letters. Thus, they are encoded with an adaptive arithmetic coder without regarding any context.

The three-valued strings σ_n are encoded with an arithmetic coder conditioned on context [17], which is collected from previously reconstructed values of $\sigma_n'[i]$. For this task, a conditioning sequence $\kappa_n[i]$ is constructed from $\sigma_n'[i]$ by thresholding

$$\kappa_n[i] = \begin{cases} -1, & \text{if } \sigma_n'[i] < 0 \\ 0, & \text{if } \sigma_n'[i] = 0 \\ +1, & \text{if } \sigma_n'[i] > 0. \end{cases} \qquad (26)$$

Denote by (j, k) the coordinates and by l the pyramid level of the pixel belonging to the index i. For convenience, we set $\kappa_n^l(j, k) = \kappa_n[i]$. Then the values of $\kappa_n^l(j-1, k)$ and $\kappa_n^l(j, k-1)$ are used to build a local conditioning context. Additionally, the significance information of $\kappa_n^{l+1}(j/2, k/2)$ (i.e., $|\kappa_n^{l+1}(j/2, k/2)|$) is used to utilize correlations of significance across scales. The implementation of the arithmetic coder follows the paper of Witten *et al.* [18].

D. Inverse Quantization at the Decoder

Until now, the pyramid encoder has been described. The decoder receives an arithmetically encoded bit stream, decodes it into a stream of symbols, and reconstructs from this symbol stream successively refined quantization intervals. For each coefficient, a value of the current interval must be chosen as the reconstruction value (inverse quantization). The reconstructed Laplacian levels are suitably expanded and summed up, giving the base and refinement layer DFD's $d^{(1)}$ and $d^{(0)}$.

There are several possibilities for designing the inverse quantization function. In Shapiro's original EZW coder, the

reconstruction values are chosen in the middle of the current interval. Since our pyramid decomposition is overcomplete, there is some redundancy in the unquantized pyramid. This means that not all possible combinations are consistent with the employed reduce and expand operators. We utilize this redundancy in the pyramid for quantization error reduction.

Theorem 1: Denote by $l_{Q'}^{(k)}$ the kth Laplacian level with independently quantized values. Then the quantized Laplacian levels

$$l_Q^{(k)} = l_{Q'}^{(k)} - E\left(R\left(l_{Q'}^{(k)}\right)\right), \qquad k = 1, \cdots, L - 2 \quad (27)$$

are, in general, closer (in the mean-squared sense) to the unquantized levels $l^{(k)}$ than the levels $l_{Q'}^{(k)}$.

Proof: For least squares pyramids, the concatenation of reduce and expand operators yields the identity operator or an orthogonal projection operator P (depending on the order)

$$R(E(x)) = x$$

and

$$P(x) := E(R(x)) = P(P(x)), \qquad \forall x. \quad (28)$$

These properties are derived in Appendix B.

All "true" Laplacian levels (i.e., the levels $l^{(k)}$, $k = 1, \cdots, L - 2$) have the form $l^{(k)} = d^{(k)} - E(R(d^{(k)})) = d^{(k)} - P(d^{(k)})$. Keeping this in mind, it is easy to see that the projection of an unquantized Laplacian level is zero

$$P(l) = E(R(l)) = 0. \quad (29)$$

Denote by $l_{Q'}$ a Laplacian level with independently quantized values. We now decompose these levels into a sum of the true Laplacian level and a quantization error

$$l_{Q'} = l + q. \quad (30)$$

Application of the projection operator P on both sides leads to

$$P(l_{Q'}) = P(q) \quad (31)$$

and

$$l_{Q'} - P(l_{Q'}) = l_{Q'} - P(q) = l + q - P(q). \quad (32)$$

Thus, application of (27) projects the quantization error q linearly into the null space of P. Hence, for the new quantization error $q - P(q)$, we have $\|q - P(q)\| \le \|q\|$ with equality if and only if q belongs to the null space of P. $\square$

To utilize the above-mentioned redundancy of the pyramid decomposition, we first choose reconstruction values independently based on the quantization intervals known to the decoder. Then we project the quantization error into the vector space of admissible Laplacian levels by applying (27).

VI. DISPLACEMENT VECTOR FIELD CODING

An efficient and elegant coding method is derived by adopting the pyramid approach for coding of displaced frame differences. Specific aspects are that vector fields need to be coded lossless, and each vector consists of two components. The concept of conditioning contexts is employed because

it is much more flexible and even more efficient than zero-tree coding [10]. The flexibility is especially important for the scalable codec.

At first, coding of the base layer motion vector field $v_n^{(1)}$ is described. Similar to the coding approach for the prediction error, both components v_x and v_y of the motion vector field $v_n^{(1)}$ are decomposed separately into a Laplacian pyramid L_x and L_y of $L_v - 1$ levels, respectively. For simplicity, only one component is mentioned in the following, where no ambiguity can occur. The levels of the pyramid are obtained by

$$l_x^{(k)} = v_x^{(k)} - E_v\left(v_x^{(k+1)}\right), \qquad k = 1, \cdots, L_v - 2$$
$$l_x^{(L_v-1)} = v_x^{(L_v-1)} \quad (33)$$

where

$$v_x^{(k)} = R_v\left(v_x^{(k-1)}\right), \qquad k = 2, \cdots, L_v - 1 \quad (34)$$

denotes the levels of the intermediate Gaussian pyramid. The reduce and expand operators $R_v(\cdot)$, $E_v(\cdot)$ need to be designed carefully to obtain optimal compression. On the one hand, a high energy concentration into the higher pyramid levels should be achieved; on the other hand, one should be able to code the remaining residuals at lower levels efficiently. The two important properties of the coding technique of significance maps are that zero coefficients can be coded efficiently, and that one large residual coefficient is more expensive to code than two small coefficients.

The operators $R_v(\cdot)$ and $E_v(\cdot)$ have a 2×2 support to obtain a quadtree-like pyramid structure. As expand operator $E_v(\cdot)$, simple repetition is employed. The nonlinear operator $R_v(\cdot)$ used for reduction termed *closest couple* provides the best performance in our experiments. To achieve that at least one coefficient of the Laplacian pyramid becomes zero, this operator outputs one element of a block of 2×2 vector components. The selection criterion maximizes the number of small coefficients by first searching for the coefficient pair which has the smallest difference. The coefficient of the pair which has the smallest difference to the remaining two values is taken as output. Since half-pel motion compensation is used, the vector field components are scaled by a factor of 2 and treated as integers. Hence, the coefficients of the Laplacian pyramid also have integer precision.

The vector fields must be coded lossless, and the integer precision allows for simple bit plane coding, which could also be interpreted as a specialized version of layered "quantization." Instead of calculating a threshold Δ as for DFD coding, the position of the most significant bit is calculated. Halving of the threshold is equivalent to switching to the next lower bit plane. Significant coefficients are the coefficients with a bit set in the currently selected bit plane. Therefore, similar to DFD coding, a significance sequence σ_n is obtained for each bit plane, which is coded employing conditioning contexts followed by adaptive arithmetic coding [18]. The formulation of the different contexts is based on the conditioning sequence $\kappa_n[i]$.

Due to the quadtree structure, one class of contexts depends only on conditioning symbols $\kappa_n[i]$ within a 2×2 block. To keep the context causal, unavailable conditioning symbols $\kappa_n[i]$ are taken from the previous layer $\kappa_{n-1}[i]$. From the

3^3 possible states, the seven most frequently occurring states (experimentally determined) are used. Since, in the most significant bit plane, only a few coefficients become significant while most coefficients become significant later, the set of contexts is switched after scanning the second bit plane. In the case that all coefficients for a block are insignificant, the corresponding conditioning symbol of the next higher level is taken into account.

Finally, a different set of contexts is defined for the top level of the pyramid, which has a low-pass characteristic and no block structure, in contrast to the band-pass characteristic of the lower levels. This set is the same as used for DFD coding, except that no information on higher levels is available.

Regarding the scalable video codec, a hierarchy of the vector fields is generated (Section III-C). The vector fields do not form a pyramid since $v^{(0)}$ and $v^{(1)}$ have the same number of vectors. For calculating the refinement level $l_x^{(0)}$ of the Laplacian pyramid in (33), (18) is directly applicable

$$l_x^{(0)} = v_x^{(1)} - 2 \cdot \left(v_x^{(1)} \right) \tag{35}$$

replacing the expand function E_v by the scaling function $\hat{E}_v$. Since the decoder knows the motion vector field resolution anyway, the refinement vector field can be reconstructed from the Laplacian "pyramid." For coding, the concept of conditioning contexts is still well suited because the symbols denoting the bit plane (quantization interval) can be calculated independently of the higher resolution levels. Moreover, the conditioning contexts for entropy coding of the symbol stream are designed such that they depend only on elements from the same level.

VII. SIMULATION RESULTS

First, the suitability of the motion estimation and compensation scheme (15) in the context of the proposed scalable coder is demonstrated. Frames 6, 11, and 16 of the sequence *foreman*[4] have been coded with a nonscalable codec as well as with a scalable codec. The base layer prediction error has been decomposed into a Gaussian pyramid of four levels. Correspondingly, in the nonscalable coder, the DFD has been decomposed into five levels. For both codecs, the highest spatial resolution is CIF, hence, the base layer (level 1) has QCIF resolution. For motion estimation and compensation, blocks of 32 × 32 are matched on the refinement level; thus, the block size on level 1 is 16 × 16. Both coders run at the same bit rate of 96 kbit/s at 5 frames/s (fps), and since the coder control assigns a constant bit budget to each frame, the coding results can be compared. The first frame (1) has been coded with 48 kbits.

In the nonscalable coder, the prediction for the "base level" is obtained by reducing the prediction of the "refinement level," according to (12). This is equivalent to a reduction of the reconstructed refinement layer to obtain the reconstructed base layer frame. The prediction gains measured in decibels for the nonscalable and the scalable codec in Table I are comparable.

[4] The sequences are obtained from the original CCIR sequences using least squares cubic spline filters.

TABLE I
PSNR IN DECIBELS OF THE RESOLUTION LEVELS AFTER MOTION COMPENSATION

Frame	Level	Nonscalable	Scalable
6	0	26.9	27.1
	1	27.7	27.9
11	0	28.7	28.5
	1	30.2	29.8
16	0	29.8	29.2
	1	31.6	30.9

The diagrams in Fig. 6 show the overall coding performance for the sequence *silent*, coded at 96 kbit/s and 5 fps with both a scalable coder as well as a nonscalable coder. The base layer DFD is decomposed into four levels. For motion compensation, blocks of 8 × 8 on the base layer and 16 × 16 on the refinement layer have been used. The pyramid for coding the base layer vector field has three levels.

The PSNR of the base layer compared to the coding performance of a nonscalable coder (QCIF) running at the same average bit rate as the scalable coder (56 kbit/s) is given in Fig. 6(a). Except for the first frames, the performance is similar, as expected. For the refinement layer, the PSNR values are shown in Fig. 6(b). The comparison with the nonscalable coder running at the same total bit rate of 96 kbit/s indicates an upper bound. Furthermore, the results are compared with the simulcast case. The average bit rate of 40 kbit/s devoted to the refinement layer in the scalable coder is used to code the CIF sequence with the nonscalable coder independently.

On hand of the following diagrams, the coder is analyzed in more detail. Regarding the rate allocation shown in Fig. 6(c), the bit rate per frame is absolutely constant, while a varying part is allocated by the base layer. The difference is the rate available for the refinement layer. With only about 40 kbit/s, the additional refinement layers can be transmitted at a sufficient quality. As an example, the coded refinement layer image 61 is shown in Fig. 7.

The diagram in Fig. 8(a) shows solely the motion compensation performance of the base layer compared to a nonscalable QCIF coder. Both coders run in a closed loop at a fixed bit rate including update coding, but only the PSNR after motion compensation before update coding is shown. As expected, the gain is similar. In Fig. 8(b), the performance of the refinement layer compared to a nonscalable CIF coder running at 96 kbit/s is given. Although the curves differ due to different bit allocations, the overall motion compensation gain is comparable.

As is shown in Fig. 9, coding the CIF sequence *container* at 48 kbit/s and 5 fps with the scalable coding scheme also results in a gain compared to the simulcast case, where the refinement level coder runs at 25 kbit/s.

VIII. CHARACTERISTICS OF THE PROPOSED CODING METHOD

The proposed scheme for a spatially scalable coder relies on a predictive motion-compensated approach. In contrast to spatiotemporal coding methods which utilize correlations of more than two frames (e.g., 3-D subband coding or motion estimation with more than one frame), such coders do not necessarily introduce a delay of more than one frame. The update information (the coded DFD) is calculated on a frame-by-
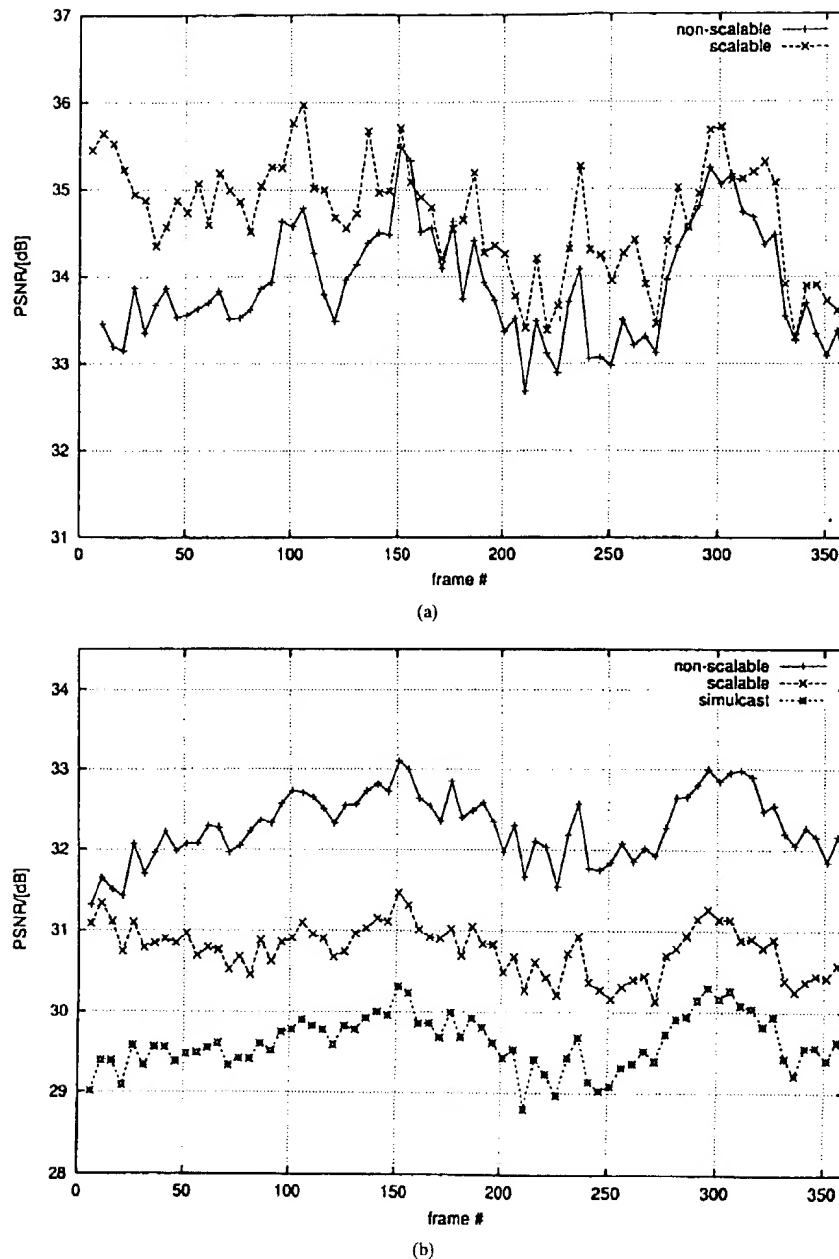
(a)

(b)

Fig. 6.   Coding performance of the scalable coder for the sequence *silent* at 5 fps. (a) PSNR of the base layer compared to the nonscalable coder running at the same bit rate (56 kbit/s). (b) PSNR of the refinement layer compared to the nonscalable coder running at the same bit rate (96 kbit/s) and the simulcast case (nonscalable CIF coder at 40 kbit/s).

frame basis, thus retaining this low-delay property. Therefore, the proposed coder is especially suited for communications applications.

One key feature is the decomposition of the displaced frame differences into a Laplacian pyramid. In connection with overlap block motion compensation, the coder produces no blocking artifacts. The artifacts are similar to the artifacts of subband coders and are dependent on the used filters.

There is no need for a special start-up procedure since rough (low-pass) approximations of large areas are possible with a constrained bit rate. On the other hand, if most parts of the

DFD are close to zero, the available bit rate can be spent in small regions, thus allowing an update of fine details. This property simplifies coder control considerably after a scene cut. Furthermore, there is no need for an explicit inter/intraswitch because the Laplacian pyramid is suited for static image statistics as well.

If only still images are transmitted (or if there is only slight motion present), the reconstructed image at the receiver will converge to a perfect reconstruction. This means that still images are automatically progressively coded (see Appendixes A–B).
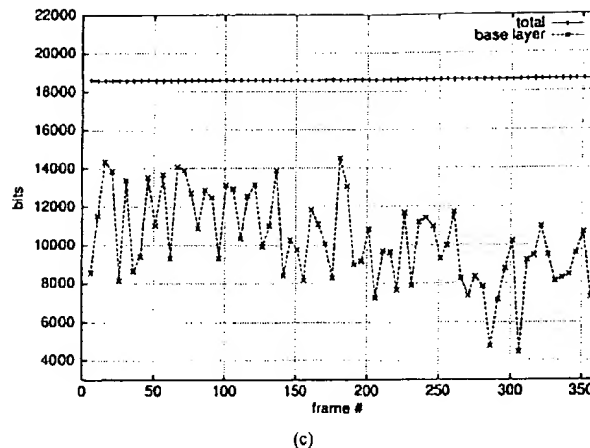
(c)

Fig. 6. (*Continued.*) Coding performance of the scalable coder for the sequence *silent* at 5 fps. (c) Rate allocation for the base layer and for the complete frame.
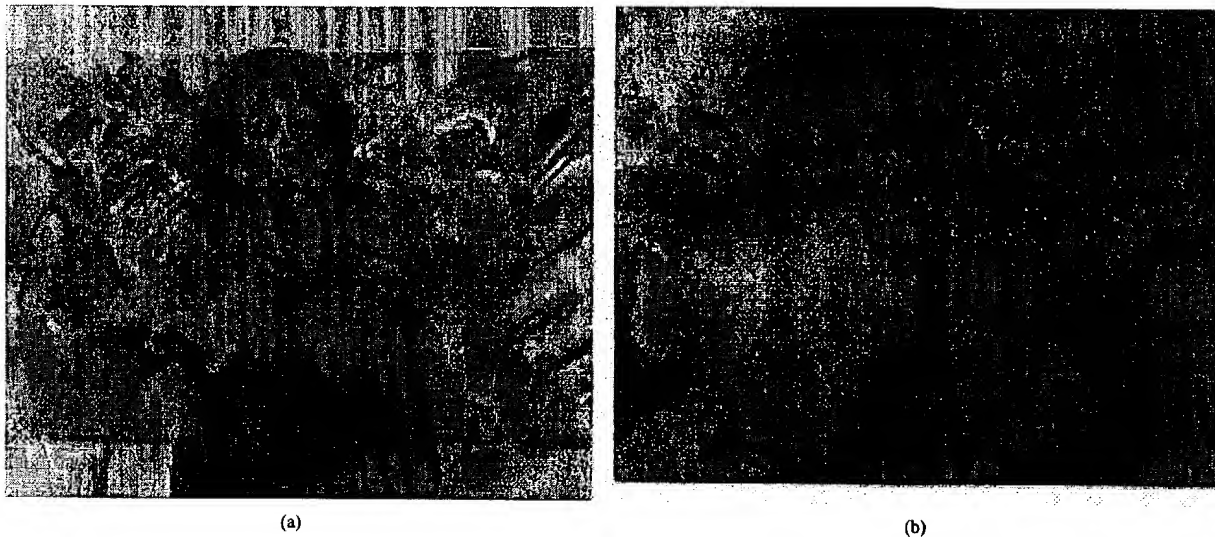

(a)

(b)

Fig. 7. Frame 61 of test sequence *silent* coded with the scalable coder (CIF, 96 kbit/s, 5 fps). (a) Coded refinement layer of frame 61 at 40 kbit/s. (b) Close up.

The DFD coder outputs an embedded bit stream, which eases coder control and adds flexibility. The sequence coder control may truncate the bit stream at *any* point (usually when the available bit budget is exhausted, or the buffer exceeds a given threshold). Thus, the coder has the ability to code frames at a *constant* bit rate, eliminating the need for a buffer, and thus reducing the delay. Since there is no separate encoding of address information and quantized data, the algorithm works efficiently in a wide range of bit rates.

IX. DISCUSSION AND SUMMARY

A new spatially scalable predictive video coding scheme is described in this paper. The basic feature is a predictive coding approach for the base layer as well as for the refinement layer, such that the displaced frame differences can be decomposed into a *single* Laplacian pyramid.

One major aspect of the paper is the specific design of motion estimation and compensation. Motion compensation is performed such that the displaced frame difference of the

base layer becomes similar to the reduced prediction error of the refinement layer. Therefore, motion is estimated and compensated on interpolated frames approximating the higher resolution level in the least square sense. As filters, centered cubic spline filters are chosen. The relation of the motion-compensated predictions on the base and the refinement layer determines the complete coder performance. Further research is directed to the improvement of the robustness when the predictions differ significantly.

Motion is estimated in a hierarchical fashion, taking into account the motion vector field of the next lower resolution level. Since the difference between levels represents an additional refinement information, a similar approach as for coding of the displaced frame differences is designed to encode the vector fields efficiently using a Laplacian pyramid coding approach employing conditioning contexts.

The second main aspect is the design of an improved Laplacian pyramid decomposition of the base layer. It turns out that centered cubic spline filters are among the best linear
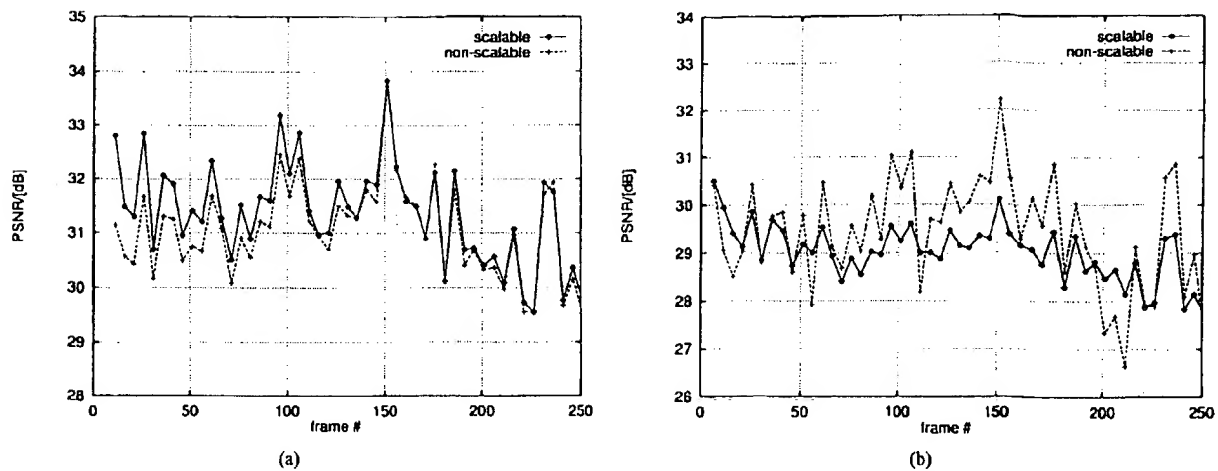
Fig. 8. Motion compensation performance. (a) Motion compensation gain of the base layer compared to a nonscalable QCIF coder. (b) Motion compensation gain of the refinement layer compared to a nonscalable CIF coder.
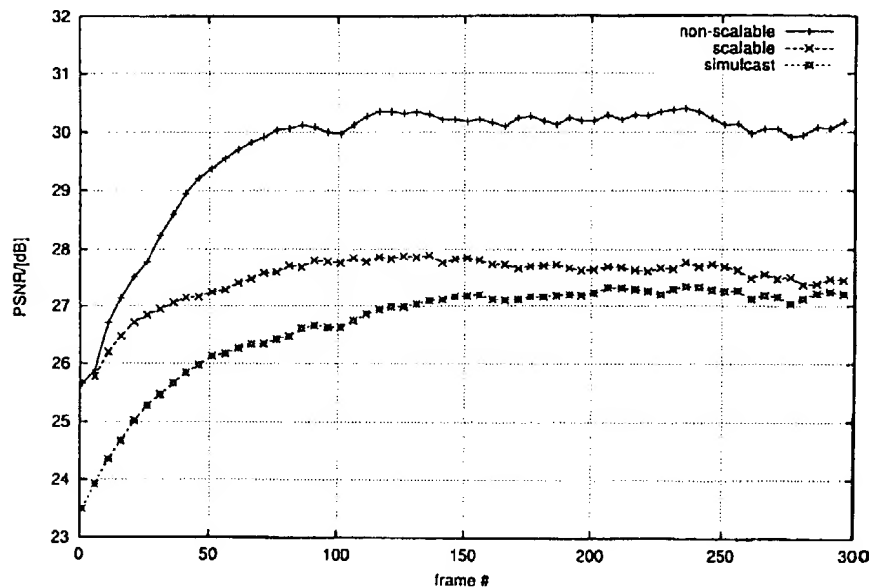

Fig. 9. Coding performance of the scalable coder for the sequence *container* at 5 fps.

filters fulfilling the desired design objectives. Interestingly, the *same* filters meet the constraints for scalable motion compensation as well as for efficient pyramid decomposition.

Furthermore, a rate-distortion efficient embedded DFD coding method using conditioning contexts for coding is developed, which allows for progressive transmission and simplifies coder control significantly. Simulation results are provided for the complete hybrid video coder.

APPENDIX A
ANALYSIS OF THE QUANTIZATION ERRORS

A. Open-Loop Quantization

The effects of quantizing the Laplacian DFD pyramid L in an open loop are analyzed in the following. In an open-loop

approach the quantization error $q^{(k)}$ at level k is independent of quantization errors at other levels $q^{(i)}, i \neq k$

$$\tilde{l}^{(k)} = l^{(k)} + q^{(k)} \tag{36}$$

The coded displaced frame difference $\tilde{d}^{(k)}$ at level k is then given by

$$\tilde{d}^{(k)} = \tilde{l}^{(k)} + E\left(\tilde{d}^{(k+1)}\right)$$

$$= l^{(k)} + q^{(k)} + \sum_{i=k+1}^{L-1} E^{i-k}\left(l^{(i)}\right) + \sum_{i=k+1}^{L-1} E^{i-k}\left(q^{(i)}\right).$$

With the convention $E^0(x) = x$, this leads to

$$\tilde{d}^{(k)} = d^{(k)} + \sum_{i=k}^{L-1} E^{i-k}\left(q^{(i)}\right). \tag{37}$$

Thus, the quantization error $q'^{(k)}$ of the displaced frame

difference at level k is given by

$$q'^{(k)} := \tilde{d}^{(k)} - d^{(k)} = \sum_{i=k}^{L-1} E^{i-k}\left(q^{(i)}\right). \tag{38}$$

Based on this relation, the quantization error for the coded frame is given by

$$\begin{aligned}
\tilde{g}^{(k)} &= \hat{g}^{(k)} + \tilde{d}^{(k)} \\
&= g^{(k)} + q'^{(k)}.
\end{aligned} \tag{39}$$

Therefore, the difference between the coded refinement image (level 0) and the coded and expanded base layer (level 1) is affected only by the quantization error of the refinement layer

$$\begin{aligned}
\tilde{g}^{(0)} - E\left(\tilde{g}^{(1)}\right) &= g^{(0)} + q'^{(0)} - E\left(g^{(1)} + q'^{(1)}\right) \\
&= g^{(0)} - E\left(g^{(1)}\right) + q^{(0)}.
\end{aligned} \tag{40}$$

B. Quantization Error Feedback

Furthermore, the quantization error feedback due to the predictive coding structure can be analyzed. Therefore, we assume no motion compensation (all displacement vectors are zero). Hence, the prediction is just the previously coded frame $\hat{g}_n^{(k)} = \tilde{g}_{n-1}^{(k)}$. The refinement level of the Laplacian pyramid to be coded

$$\begin{aligned}
l_n^{(0)} &= d_n^{(0)} - E\left(d_n^{(1)}\right) \\
&= g_n^{(0)} - \hat{g}_n^{(0)} - \left(E\left(g_n^{(1)}\right) - E\left(\hat{g}_n^{(1)}\right)\right) \\
&= g_n^{(0)} - \tilde{g}_{n-1}^{(0)} - \left(E\left(g_n^{(1)}\right) - E\left(\tilde{g}_{n-1}^{(1)}\right)\right)
\end{aligned}$$

depends on the reconstructed previous base and refinement level. Using (40) reveals

$$l_n^{(0)} = g_n^{(0)} - E\left(g_n^{(1)}\right) - \left(g_{n-1}^{(0)} - E\left(g_{n-1}^{(1)}\right)\right) - q_{n-1}^{(0)} \tag{41}$$

that $l_n^{(0)}$ is affected only by the quantization error $q_{n-1}^{(0)}$ of this level. No shift of quantization errors across scales occurs. Therefore, coding of still images and unchanged regions between subsequent frames converges to lossless coding.

APPENDIX B
DERIVATION OF THE REDUCE AND EXPAND OPERATORS

In this Appendix, reduce and expand operators are derived, which are based on spline approximations. The derivation follows in its main aspects [15], [9], and is given in terms of discrete one-dimensional operators. For construction of the pyramids used for motion estimation and for DFD coding, these operators are extended to the 2-D case by separable application along the rows and columns of the frames.

The main difference between [9] and our approach is the centering of the lower resolution grid with respect to the high-resolution grid. This yields several advantages compared to the classical approach. First, the centering leads to a quadtree-like topology of the pyramid. Thus, each pixel on a coarse pyramid level belongs to four corresponding pixels of the next finer level, which is advantageous for the propagation of block-based motion vector fields across the differing resolutions. Second, consistent boundary conditions are obtained by mirrored extension of the finite signals, while the uncentered

pyramid requires periodic extension which degrades the coding performance at signal support boundaries.

A. Discrete Least Squares Approximation

We consider the construction of a dyadic pyramid in a purely discrete framework. The basic operation is the approximation from a fine space S_1 onto a coarse space S_2 at twice the scale. For convenience, we focus first on sequences of infinite length. Implications of the finite support of the signals are discussed in the following subsection. Thus, we set $S_1 = l_2$ (the space of squared summable sequences), and consider the coarser subspace $S_2 \subset l_2$ generated from the even integer translates of a generating sequence h:

$$\begin{aligned}
S_2 &= \text{span}\{h[k - 2l]\}_{l\in\mathbb{Z}} \\
&= \left\{\tilde{s}_1 | \tilde{s}_1[k] = \sum_{k\in\mathbb{Z}} s_2[l]h[k - 2l], s_2 \in l_2\right\}. \tag{42}
\end{aligned}$$

We may think of $s_2[k]$ as being samples of some pyramid level. Then $\tilde{s}_1[k]$ are the samples of the corresponding interpolated (expanded) level. We therefore define the (one-dimensional) expand operator E as

$$E : l_2 \to l_2; \quad s_2 \mapsto E(s_2) = [s_2]_{\uparrow 2} * h \tag{43}$$

and S_2 becomes simply the image of E

$$S_2 = \{\tilde{s}_1 | \tilde{s}_1 = E(s_2), s_2 \in l_2\}. \tag{44}$$

Note that the generating sequence h has a twofold meaning here: 1) it generates the Riesz basis $\{h[k - 2l]\}_{l\in\mathbb{Z}}$ of S_2, and hence serves as a scaling function, and 2) it equals the impulse response of the synthesis filter, thus defining the expand operator.

The reduce operator R consists of an antialiasing filter (or analysis filter) with impulse response $\overset{\circ}{h}$ followed by a subsampler:

$$R : l_2 \to l_2; \quad s_1 \mapsto s_2 = R(s_1) = \left[s_1 * \overset{\circ}{h}\right]_{\downarrow 2}. \tag{45}$$

The concatenation of R and E approximates any sequence $s_1 \in l_2$ with a sequence $\tilde{s}_1 = E(R(s_1)) \in S_2$. A prefilter $\overset{\circ}{h}$ is called *optimal* in the least squares sense with respect to a synthesis filter h when the energy $\sum_{k\in\mathbb{Z}}(s_1[k] - \tilde{s}_1[k])^2$ of the approximation error becomes minimal.

Theorem 2: The optimal prefilter with respect to a synthesis filter with z transform $H(z)$ is given by

$$\overset{\circ}{H}(z) = \frac{2H(z^{-1})}{H(z)H(z^{-1}) + H(-z)H(-z^{-1})}. \tag{46}$$

Proof: The least squares approximation is achieved when the error $s_1[k] - \tilde{s}_1[k]$ is orthogonal to S_2 or, equivalently

$$\langle h[k - 2l], s_1[k]\rangle = \left\langle h[k - 2l], \sum_{n\in\mathbb{Z}} s_2[n]h[k - 2n]\right\rangle,$$
$$\forall l \in \mathbb{Z}.$$

With $h^T[k] = h[-k]$, we can rewrite the inner products as convolutions

$$[h^T * s_1]_{\downarrow 2} = [h^T * h]_{\downarrow 2} * s_2.$$

Application of the inverse of $[h^T * h]_{\downarrow 2}$ on both sides (which exists because we require h constituting a Riesz basis) yields

$$s_2 = ([h^T * h]_{\downarrow 2})^{-1} * [h^T * s_1]_{\downarrow 2}$$

$$= \left[\left[((h^T * h]_{\downarrow 2})^{-1} \right]_{\uparrow 2} * h^T * s_1 \right]_{\downarrow 2}$$

$$= \left[\overset{\circ}{h} * s_1 \right]_{\downarrow 2}.$$

Thus, the optimal prefilter has the impulse response $\overset{\circ}{h} = [((h^T * h]_{\downarrow 2})^{-1}]_{\uparrow 2} * h^T$, whose z transform is given by (46). ∎

Theorem 3: Least squares pyramids have the property that reduction of an expanded signal leaves the signal unchanged.

Proof: By definition, the reduce operator R of a least squares pyramid is chosen such that the energy of the approximation error is minimized. This being true, the concatenated operator

$$P : l_2 \to S_2, \quad s \mapsto \tilde{s} = E(R(s))$$

must be an orthogonal projector from l_2 into S_2. This implies $P(P(s)) = P(s)$. For the reduce and expand operators, this means that $E(R(E(R(s)))) = E(R(s))$, and

$$R(E(s)) = s, \qquad \forall s \in l_2 \tag{47}$$

follows immediately. ∎

This property is crucial for the scalable coding scheme described in this paper.

B. Centered Spline Approximation

Until now, we have not specified the space S_2 and the filter h. We confine ourselves to spline pyramids mainly for to reasons. First, splines have excellent approximation properties [19], [20]. Second, the spline framework allows for a progressive transition between the piecewise constant and the band-limited model. The spline of degree zero leads to the Haar wavelet (best time localization), and the cardinal spline filters converge to the sinc interpolator for high degrees [21]. So it can be expected that a reasonable tradeoff between time localization and aliasing can be found inside the class of spline filters.

Usually, the generating kernel h is symmetric, and the coarser grid points are positioned on the even integers. We propose instead to shift the sampling grid of the approximating function such that the samples "sit" centered between the samples of the high-resolution grid. Using the formalism developed in [15], it is not difficult to derive the z transform $H_\Delta(z)$ of the cardinal spline interpolator corresponding to a shift Δ

$$H_{\Delta(z)} = \frac{B^n_{2,\Delta}(z)}{B^n_{1,0}(z)} \tag{48}$$

where the sampled B-spline filter kernels $B^n_{m,\Delta}(z)$ are defined as

$$B^n_{m,\Delta}(z) = \sum_{k \in \mathbb{Z}} \beta^n \left(\frac{k - \Delta}{m} \right) z^{-k} \tag{49}$$
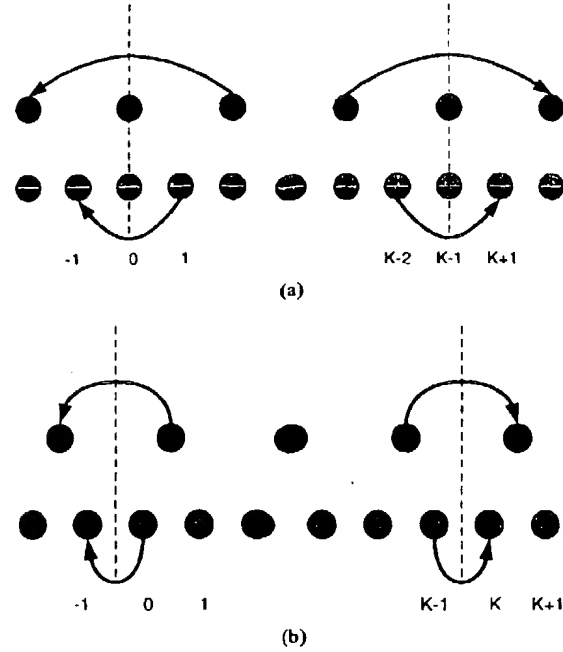

Fig. 10. Odd and even mirroring of a discrete signal. (a) Odd mirroring in case of a traditional pyramid. (b) Even mirroring in case of a centered pyramid.

and $\beta^n(x)$ is the symmetrical B spline of degree n, defined by

$$\beta^n(x) = \beta^0(x) * \beta^{n-1}(x)$$

with

$$\beta^0(x) = \text{rect}(x) = \begin{cases} 1, & \text{for } |x| \leq 1/2 \\ 0, & \text{for } |x| > 1/2. \end{cases}$$

For the shift parameter $\Delta = 0$, the pyramid equals the one derived in [9], and corresponds to the traditional formulation of the discrete wavelet transform. For $\Delta = 1$, the samples of the coarse grid sit on the odd integers, and for $\Delta = 1/2$, we end up with the *centered* pyramid which we use for the compression algorithm. In Fig. 10, the positioning of the sampling grids is depicted for the traditional ($\Delta = 0$) and the centered ($\Delta = 1/2$) case. The centered pyramid allows us to assign each sample on the fine grid the closest sample on the coarse grid, whereas this is possible in the traditional pyramid only for the even samples.

Until now, only infinite signals have been considered. In the case of finite signals, we have to impose suited boundary conditions. In signal processing, such boundary conditions are usually formulated in terms of signal extensions across the boundaries. It is easy to show that in the case of the uncentered spline approximation, consistent boundary conditions are given by periodic repetition of the signals. However, this introduces unnatural discontinuities at the boundaries, which degrade the performance of most signal processing applications. Therefore, a symmetric extension by mirroring is preferred in many applications, as depicted in Fig. 11 for continuous signals.

If the length of the signal K is even, a proper symmetric extension is impossible as long as we stay with the uncentered grid. On the other hand, if we use a centered grid as proposed

Fig. 11. Symmetric boundary extension of a continuous signal with finite support.

in this paper, we can use the "even" mirroring

$$s[-k - 1] = s[k], \qquad k = 0 \cdots K - 1$$
$$s[k] = s[2K - 1 - k], \qquad k = K \cdots 2K - 1$$

without distorting property (47).

ACKNOWLEDGMENT

The authors would like to thank Prof. Dr.-Ing. H. D. Lüke, Aachen University of Technology, Germany, for his support of the underlying research, and also the anonymous referees for their valuable comments.

REFERENCES

[1] T. Hanamura, W. Kameyama, and H. Tominaga, "Hierarchical coding scheme of video signals with scalability and compatibility," *Signal Processing: Image Commun.*, vol. 5, pp. 159–184, Feb. 1993.

[2] B. Girod, U. Horn, and B. Belzer, "Scalable video coding with multiscale motion compensation and unequal error protection," in *Proc. Symp. Multimedia Communication Video Coding*, New York, NY, Oct. 1995.

[3] P.-Y. Cheng, J. Li, and C.-C. Jay Kuo, "Multiscale video compression using wavelet transform and motion compensation," in *Proc. IEEE Int. Conf. Image Processing*, vol. I, Washington DC, 1995, pp. 606–609.

[4] T. Naveen and J. W. Woods, "Motion compensated multiresolution transmission of high definition video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 29–41, Feb. 1994.

[5] K. Tsunashima, J. B. Stampleman, and V. M. Bove, "A scalable motion-compensated subband image coder," *IEEE Trans. Commun.*, vol. 42, pp. 1894–1901, Feb./Mar./Apr. 1994.

[6] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[7] D. Taubmann and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept. 1994.

[8] F. Müller, K. Illgner, and B. Menser, "Embedded Laplacian pyramid image coding using conditional arithmetic coding," in *Proc. IEEE Int. Conf. Image Processing, ICIP'96*, vol. I, Lausanne, Switzerland, Sept. 1996, pp. 221–224.

[9] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II—Efficient design and applications," *IEEE Trans. Signal Processing*, vol. 41, pp. 834–848, May 1993.

[10] K. Illgner and F. Müller, "Hierarchical coding of motion vector fields," in *Proc. IEEE Int. Conf. Image Processing, ICIP'95*, vol. I, Washington DC, Oct. 1995, pp. 566–569.

[11] ITU-T International Telecommunication Union, *Draft ITU-T Recommendation H.263 (Video Coding for Low Bitrate Communication)*, KPN Research, The Netherlands, Jan. 1995.

[12] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, Apr. 1983.

[13] O. Werner, "Drift analysis and drift reduction for multiresolution video coding," *Signal Processing: Image Commun.*, vol. 8, pp. 387–409, May 1996.

[14] B. Girod, "Motion compensation: Visual aspects, accuracy, and fundamental limits," in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds. Boston: Kluwer Academic, 1993, pp. 125–152.

[15] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I—Theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 821–833, May 1993.

[16] K. Illgner and F. Müller, "Motion estimation using overlapped block motion compensation and Gibbs-modeled vector fields," in *Proc. 9th Workshop Image and Multidimensional Signal Processing (IMDSP'96)*, Belize City, Belize, Mar. 1996, pp. 126–127.

[17] J. Rissanen and G. G. Langdon, "Universal modeling and coding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 12–23, Jan. 1981.

[18] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.

[19] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

[20] M. Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Trans. Signal Processing*, vol. 44, pp. 519–527, Mar. 1996.

[21] A. Aldroubi, M. Unser, and M. Eden, "Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Signal Processing*, vol. 28, pp. 127–138, 1992.

Klaus Illgner received the Dipl. degree (M.S.) in electrical engineering from Aachen University of Technology (RWTH), Germany, in 1991.

Since then, he has been with the Institut für Elektrische Nachrichtentechnik, RWTH Aachen, as a Scientific Assistant. His research interests cover all aspects of low-bit-rate video coding and video communications, where his main focus at present is on motion compensation and multiresolution video.

Mr. Illgner is a member of the IEEE Signal Processing Society, the IEEE Communications Society, and the German Electrical Engineering Association (VDE).

Frank Müller received the Dipl. degree (M.S.) in electrical engineering from Aachen University of Technology (RWTH), Germany, in 1991.

Since then, he has been with the Institut für Elektrische Nachrichtentechnik, RWTH Aachen, as a Scientific Assistant. His current interests are hierarchical and multiresolution approaches to image and video compression, and source coding theory.

Mr. Müller is a member of the IEEE Signal Processing Society, the IEEE Information Theory Society, the German Electrical Engineering Association (VDE), and the German Computer Society (Gesellschaft für Informatik (GI)).

AN EQUAL OPPORTUNITY EMPLOYER

Organization _____ TC2600 _____ Bldg./RoomPK2 _____

U. S. DEPARTMENT OF COMMERCE

COMMISSIONER FOR PATENTS

P.O. BOX 1450

ALEXANDRIA, VA 22313-1450

IF UNDELIVERABLE RETURN IN TEN DAYS

OFFICIAL BUSINESS

UNDELIVERABLE AS ADDRESSED-
FORWARDING ORDER EXPIRED